

TECHNICKÁ UNIVERSITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2009

Vítězslav Studnička

TECHNICKÁ UNIVERSITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B 2612 – Elektronika a informatika
Studijní obor: 1802R022 - Informatika a Logistika

Translátor

Translator

Bakalářská práce

Autor: **Vítězslav Studnička**
Vedoucí práce: Ing. Miloš Hernych
Konzultant: doc. Ing. Jiřina Královcová, Ph.D.

V Liberci 24. 5 2009

Zde se nachází originál zadání BP

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem

Datum

Podpis

Poděkování

Tímto děkuji svému vedoucímu bakalářské práce Ing. Miloši Hernychovi za jeho odbornou pomoc, věnovaný čas a vstřícnost, které mi věnoval při vytváření mé bakalářské práce. Tyto znalosti mi byly velikou oporou při řešení komplikací, zejména při řešení komplikací s EPSNet protokolem a práce se sériovým portem. Také chci poděkovat doc. Ing. Jiřině Královcové, Ph.D. za poskytnutí odborných konzultací.

Abstrakt

Cílem této bakalářské práce bylo vyvinout program Translátor pod systémem Windows nebo Linux, který bude zajišťovat komunikaci mezi PC na jedné straně a PLC na straně druhé. Program by měl umožňovat dynamické přidělování specifických automatů k různým uživatelům přes TCP/IP. První kapitola pojednává o způsobu komunikace mezi automaty a popisuje protokol, skrz který mezi sebou přenášejí informace. Rovněž popisuje jeho chování a nutnosti nastavení na nižší vrstvě sítě. Druhá kapitola popisuje architekturu TCP/IP, zejména protokol UDP. Dále je zde nahlédnuto do fungování jednotlivých vrstev přenášení dat po internetu. Třetí kapitola pojednává o sériovém portu a jeho nastavení. Čtvrtá kapitola je věnována vlastnímu popisu programu a zkušenostem získaných z praktického nasazení v laboratoři TK3 na reálných zařízeních.

Klíčová slova

PLC
EPSNet
TCP/IP
UDP
Sériový port

Abstract

The Purpose of this bachelor study is to develop a program under MS Windows or Linux system, which can provide communication between PC on one hand and PLC on the other side. Program is supposed to provide dynamic assignment of specific automatic machine to different users over TCP/IP. First chapter is discoursing about concept of communication between automatic machines and is describing protocol through informations are carried. Also describes its behaviour and need of configuration at lower net layers. Second chapter is concerned with architecture of TCP/IP, especially about UDP protocol. Further, there are parts talking about separate layers functionality and their effect on data transmitting. Third chapter is about serial port and it's specific settings. Fourth chapter paid attention to characterization of program itself and experience gained from its practical use on real machines in TK3 laboratory.

Keywords

PLC
EPSNet
TCP/IP
UDP
Serial Port

Obsah

Základní pojmy.....	8
Úvod.....	9
1.EPSNet Sít'.....	11
1.1 Nastavení sítě EPSNet.....	11
1.1.1 Nastavení sítě typu monomaster.....	11
1.1.2 Nastavení sítě typu multimaster.....	12
1.1.3 Nastavení sítě typu EPSNet-F.....	12
1.2 Typy komunikace v síti EPSNet.....	12
1.3 Zahájení komunikace a identifikace subjektu:.....	13
1.4 Provozní funkce.....	14
1.4.1 Systémové provozní funkce.....	14
1.4.2 Veřejné provozní funkce.....	14
1.4.3 Veřejné datové služby	15
1.5 Kontrola chyb v komunikaci.....	15
1.5.1 Kontrola sudou paritou.....	16
1.5.2 Kontrola pomocí CRC (Cyclic redundancy check).....	16
1.6 Adresování v Ethernetu.....	16
1.7 Protokol EPSNet UDP.....	17
1.7.1 EPSNet UDP datagram.....	17
1.8 Komunikace PC a PLC v odlišných sítích.....	18
2. Protokol TCP/IP.....	20
2.1 Protokol UDP – User Datagram Protocol.....	21
2.1.1 Hlavička UDP.....	23
2.2 IP – InterNet Protocol.....	23
2.2.1 IP datagram.....	24
2.3 Vrstvení EPSNet zprávy	24
3. Sériový port RS-232.....	26
3.1 Parita.....	27
3.2 Baudrate.....	27
3.3 Stop bit.....	28
4. Program Translátor.....	29
4.1 Princip fungování programu.....	29
4.2 Translátor: směrovací aplikace.....	29
4.3 Translátor: klient.....	30
4.4 Spuštění Translátoru a vytvoření spojení.....	31
4.5 Průběh vývoje.....	32
4.6 Zásady používání.....	33
4.6.1 Diagram vytvořené sítě.....	33
Závěr.....	34
Příloha A.....	35
Příloha B.....	36
Seznam použité literatury.....	37

Základní pojmy

Delphi	... integrované vývojové prostředí firmy Borland pro systémy MS Windows v jazyce Object Pascal
EPSNet	... protokol firmy Teco a.s. využívaný k síťové komunikaci mezi PLC
IP	... protokol určený k přenosu dat mezi počítači připojených k Internetu
INDY	... balíček síťových komponent pro vývojové prostředí Delphi
Mosaic	... integrované vývojové prostředí firmy Teco a.s.
PLC	... programovatelný logický automat používaný pro automatizaci procesů v reálném čase
Sériový port	... hardwarové datové rozhraní k zapojení elektronických přístrojů do počítače nebo k jinému počítači
TCP/IP	... skupina protokolů pro komunikaci v počítačové síti
UDP	... protokol transportní vrstvy v síti TCP/IP

Úvod

Programovatelné automaty PLC jsou mikropočítačová zařízení se vstupy a výstupy schopnými ovládat velkou škálu dynamických systémů. Nácházejí využití nejenom v průmyslu, kde mohou řídit výrobní linky, průmyslové roboty a manipulátory, ale i další systémy. Dnes nacházejí uplatnění i v aplikacích terciální sféry – automatizace budov, řízení vytápění a klimatizace, v dopravě a podobně.

Starší systémy PLC Tecomat řady TC500 a TC600, které jsou využívány pro řízení fyzikálních modelů v laboratoři TK3, jsou pro komunikaci s vývojovým prostředím Mosaic, používaným pro ladění řídicích aplikací v těchto automatech, vybaveny rozhraním RS232. Ladění aplikací je tak možné pouze z počítače, bezprostředně připojeného k danému automatu. Cílem této bakalářské práce bylo vytvořit sadu programů, které umožní přístup k daným automatům z libovolného počítače, připojeného k Ethernetu v laboratoři TK3. Programy musí zajistit kontinuální výměnu dat mezi počítači s přímým propojením s PLC a počítači s aktivním vývojovým prostředím Mosaic a také možnost dynamického párování počítačů, které budou takto vzájemně propojeny.

Prostředí Mosaic s PLC Tecomat komunikuje protokolem EPSNet, který existuje ve variantě pro přenos po sériové lince a Ethernetu. Hlavní myšlenkou vyvíjeného nástroje bylo „překládat“ datagramy sériové varianty EPSNetu na variantu Ethernetovou a zpět. Z tohoto důvodu byl nástroj nazván Translátor.

Funkcí jedné části programu Translátor je v počítači připojeném k fyzikálnímu modelu zachytávat datagramy přicházející z PLC Tecomat, transformovat je do verze vhodné pro přenos po Ethernetu a odesílat na počítač, ve kterém běží druhá část Translátoru, zajišťující spojení se vzdáleným počítačem, na kterém běží vývojové prostředí Mosaic. Naopak datagramy z Mosaicu jsou stejnou cestou přenášeny zpět, transformovány a odesílány do PLC.

Pro tvorbu a provoz programu bylo vybráno prostředí MS Windows, jelikož všechny počítače, které přicházejí do úvahy, že by mohly využívat tento software, pracují s tímto operačním systémem. Za vývojové prostředí bylo vybráno Delphi 2007, nativně podporující komunikaci přes TCP/IP pomocí komponent.

V první kapitole je podrobně analyzován způsob komunikace a protokol využívaný automaty od firmy Teco. Jsou zde rozebrány důležité systémové zprávy včetně nastavení zařízení pro správnou funkčnost. V kapitole druhé je popsána architektura a metoda přenosu dat po síti TCP/IP. Třetí kapitola obsahuje informace o sériovém portu a jeho konfiguraci pro vznik spojení. Poslední kapitola se věnuje vlastnímu praktickému využití předchozích kapitol při vývoji programu Translátor včetně principu fungování a návodu pro používání.

1. EPSNet Sít'

Protokol EPSNet je využíván v PLC od firmy Teco a.s. pro komunikaci jak mezi zařízeními PLC a PLC, tak i v komunikaci mezi PC a PLC. Znalost tohoto protokolu, zejména pak EPSNet UDP, je třeba ke zpracování dat posílaných z a od PLC v programu Translátor viz kapitola 4.

V síti pracují dva druhy účastníků, nadřízený a podřízený. Komunikace je založena na principu dotaz a odpověď. Komunikace je jednostranně orientována, tzn. že dotazovat se může pouze nadřízená stanice.

1. **stanice master:** Nadřízená stanice. Vysílá dotaz směrem k podřízené stanici.
2. **stanice slave:** Podřízená stanice. Přijímá dotaz a odesílá odpovědi. Sama dotazy nevysílá.

V jednom momentu může podřízená stanice komunikovat pouze s jednou nadřízenou stanicí. Podřízené stanice spolu komunikovat nemohou.

1.1 Nastavení sítě EPSNet

Sít' lze projektovat třemi možnými způsoby, přičemž volba se řídí účelem a strukturou komunikace.

1. **typ monomaster:** Sít' se skládá z jedné stanice master a jedné nebo více stanic slave.
2. **typ multimaster:** Sít' se skládá z více stanic master a jedné nebo více stanic slave.
3. **typ EPSNet-F:** Sít' se skládá pouze ze stanic master a žádné stanice slave.

1.1.1 Nastavení sítě typu monomaster

Sít' obsahuje jednu stanici master a ji podřízené stanice slave. Maximální počet podřízených stanic je omezen číslem 126, ovšem reálný počet určuje typ a technická způsobilost přenosového média. Celkový počet stanic včetně nadřízené je 127.

Provoz v síti je řízen pouze danou stanicí master, z pravidla je to počítač s ovládacím softwarem. Výměna dat mezi podřízenými stanicemi je možná pouze přes nadřízenou stanicí. Přímá výměna dat mezi podřízenými stanicemi není umožněna vzhledem k principu fungování sítě.

1.1.2 Nastavení sítě typu multimaster

Síť obsahuje více stanic master a jim podřízené stanice slave. Maximální počet stanic v síti je 127 kde počet nadřízených stanic není jiným způsobem omezen. Provoz v síti je řízen vždy pouze jednou nadřízenou stanicí v daný okamžik. Během této doby se všechny ostatní stanice včetně jiných nadřízených stanic chovají jako podřízené. Po skončení komunikace této nadřízené stanice se předá řízení následující nadřízené stanici.

Výměna dat mezi dvěma nadřízenými stanicemi funguje jako mezi nadřízenou a podřízenou. Přímá výměna dat mezi podřízenými stanicemi není umožněna vzhledem k principu fungování sítě.

1.1.3 Nastavení sítě typu EPSNet-F

Síť obsahuje pouze stanice master. Slouží k rychlé výměně dat mezi stanicemi, aktuálně nadřízená stanice v jednom okamžiku data vysílá a všechny ostatní je přijímají. Lze ji využít pro samostatné spojení mezi nadřízenými stanicemi v jiných sítích, jelikož je tato komunikace na jiných kanálech na sobě nezávislá.

1.2 Typy komunikace v síti EPSNet

Protokol využívá dvou typů dotazů. Systémové spojení, to může být v jednu chvíli aktivní pouze jedno v daném kanále a dále datové spojení, které slouží pro vlastní přenos dat.

V Tab. 2.2.1 lze vidět označení částí EPSNet zprávy, zejména pak důležité jsou symboly SD1, SD2 a ED s jejich konstantními hodnotami. Analyzování těchto struktur bylo využíváno při vytváření fiktivního spojení s PLC a testování komunikace.

Tab. 1.2.1: Označení symbolů v EPSNet paketu

DA	Cílová adresa (Destination address) hodnota od 0 do 126
DATA	Samotná zpráva, data velikost maximálně 246 bytů
ED	Závěrečný znak (end delimiter) konstantní hodnota \$16
FC	Řídící byte rámce (frame control byte)
FCS	Kontrolní součet rámce (frame check sum)
LE	Délka dat (length)
LER	Opakování délky dat (length repeat)
SA	Adresa odesílatele (source address) hodnota od 0 do 126
SAC	Krátké potvrzení (short acknowledge) konstantní hodnota \$E5
SD1	Počáteční znak 1 (start delimiter 1) konstantní hodnota \$10
SD2	Počáteční znak 2 (start delimiter 2) konstantní hodnota \$68
SD4	Počáteční znak 4 (start delimiter 4) konstantní hodnota \$DC
SD2R	Opakování počátečního znaku 2 (start delimiter 2 repeat) konstantní hodnota \$68

1.3 Zahájení komunikace a identifikace subjektu:

1. Funkce **CONNECT**: Inicializuje spojení s dotazovaným subjektem. Tato funkce byla přímo využívána při testování spojení s Translátorem.

Struktura paketu od nadřazené stanice:

SD1	DA	SA	FC	FCS	ED
\$10	\$00	\$01	\$69	\$6A	\$16

FC = \$49 nebo \$69 => inicializace komunikace

Struktura paketu od podřízené stanice:

SD1	DA	SA	FC	FCS	ED
\$10	\$01	\$00	\$00	\$01	\$16

FC = \$00 => potvrzení komunikace

DA = adresa subjektu kterému je paket určen

SA = adresa subjektu v síti který vysílá paket

FCS = poslední byte hexadecimálního součtu od DA až po FC

Zde jsou adresy nadřazené stanice 00 a podřazené stanice 01

2. Funkce **IDENT**: Zjišťuje informace o dotazovaném subjektu jako je typ PLC, verze softwaru a další.

Struktura paketu od nadřazené stanice:

SD1	DA	SA	FC	FCS	ED
\$10	\$00	\$01	\$69	\$6A	\$16

FC = \$4E nebo \$6E => zažádání informací o systému

Struktura paketu od podřazené stanice:

SD2	LE	LER	SD2R	DA	SA	FC	DATA	FCS	ED
\$68	\$XX	\$XX	\$68	\$01	\$00	\$00	\$XX \$XX ...	\$XX	\$16

FC = \$00 => potvrzení

1.4 Provozní funkce

Dělí se na dva typy podle možnosti přístupu.

1.4.1 Systémové provozní funkce

Využívají se k ladění a programování systému např. prostředím Mosaic. V jednom momentě je lze používat pouze na jednom komunikačním kanálu podporujícím systémové zprávy. Systémové kanály jsou kanály probíhající v režimu **PC**.

1. Funkce **SETCW**: Nastavení řídicího slova (Set Control Word).
2. Funkce **MASKCW**: Nastavení bitů řídicího slova.

1.4.2 Veřejné provozní funkce

1. Funkce **GETCW**: Načtení stavového slova (Get State Word).
2. Funkce **GETRR**: Načtení chyb ze zásobníku.

1.4.3 Veřejné datové služby

Slouží k vlastní výměně dat mezi subjekty. Tyto služby jsou dostupné na všech kanálech probíhajících v režimu **PC** nebo **MPC**.

Tab. 1.4.3.1: Seznam veřejných datových služeb

READN	čtení obsahů registrů zápisníkové paměti a obsahu přídatné paměti dat DataBox
WRITEN	zápis do registrů zápisníkové paměti a obsahu přídatné paměti dat DataBox
WANDRN	zápis do a čtení obsahu registrů zápisníkové paměti a obsahu přídatné paměti dat DataBox
READB	čtení obsahu bitů zápisníkové paměti
WRITEB	zápis do bitů v zápisníkové paměti
READBD	přečtení obsahu bitů zápisníkové paměti a jejich následné vynulování
READND	čtení obsahu registrů zápisníkové paměti, resp. obsahu přídatné paměti dat DataBox a jejich následné vynulování
WANDRN	zápis do registrů a čtení obsahu registrů zápisníkové paměti, resp. obsahu přídatné paměti dat DataBox a jejich následné vynulování

1.5 Kontrola chyb v komunikaci

Při přenosu dat může dojít k chybám způsobených závadou nebo vnějšími vlivy. Pokud dojde k přerušení linky a počítač přestane dostávat data, chyba se odhalí sama. Horší situace nastane, pokud počítač omylem přijme log.0 místo log.1. Počítač nepozná nastalou chybu a dále pracuje se špatnými daty. Z tohoto důvodu se používá různých zabezpečovacích prostředků. Jednou z metod zabezpečení je použití zabezpečovacích kódů. K datům samotným se přidá další informace umožňující příjemci zjistit, zda jsou došla data v pořádku. Tato přidaná informace samozřejmě prodlužuje délku přenosu.

Protokol EPSNet využívá dvou druhů zabezpečení:

1. Kontrola sudou paritou
2. Kontrola CRC (cyklický redundantní kód)

1.5.1 Kontrola sudou paritou

Patří k nejjednodušším způsobům detekce chyb. Dokáže chyby pouze nalézt. Princip sudé parity je takový, že k přenášeným datům přidáme jeden bit navíc (tzv. paritní bit). O tom, zda bude paritní bit 0 nebo 1 rozhoduje celkový počet log.1 ve zprávě. Pokud bude počet 1 lichý, paritní bit bude 1, tím se doplní počet 1 na sudý. Odtud vyplývá název sudá parita. Pokud bude počet 1 sudý bude paritní bit 0 a počet se nemění. Při obdržení vadné zprávy ji systém zahodí a dále nezpracovává, tím pádem nedochází k odpovědi a nadřízená stanice opakuje dotaz.

1.5.2 Kontrola pomocí CRC (Cyclic redundancy check)

CRC je vhodný pro zjišťování chyb vzniklých v důsledku selhání techniky. Schopnost detekce chyb závisí na volbě klíče. Při správné volbě hodnoty mají delší klíče lepší schopnost detekce chyb. Protože CRC metoda je založena na dělení, nerozezná přidané nuly na začátku vstupních dat. Proto se někdy při výpočtu CRC před vstupní data přidává log. 1. Číslo za písmeny CRC určuje stupeň řídicího polynomu, např. CRC16 je kontrolní součet typu CRC s řídicím polynomem stupně 16 využívaný protokolem EPSNet. Ke kontrole slouží systémová instrukce CRCM a způsob výpočtu je $x^{16} + x^{15} + x^2 + 1$. Stupeň polynomu určuje počet bitů využívaných ke kontrole CRC. Takto vypočtená hodnota je vložena na konec přenášeného bloku. Pokud vychází $A0 = 0$ jsou data platná.

1.6 Adresování v Ethernetu

Každý kanál Ethernet má svoji IP adresu a IP masku. Jejich hodnoty závisí na nastavení protějšího účastníka komunikace. Obecně platí zásada, že IP adresy obou účastníků komunikace musí být shodné v těch místech, kde má IP maska nenulovou hodnotu. IP maska by měla být pro oba účastníky shodná, není to však podmínkou.

Tab. 1.6.1: Ukázka nastavení lokální sítě

PC		PLC	
IP adresa:	192.168.1.1	IP adresa:	192.168.1.2
IP maska:	255.255.255.0	IP maska:	255.255.255.0

Tab. 1.6.2: Ukázka nastavení lokální sítě 2

PC		PLC	
IP adresa:	192.168.12.1	IP adresa:	192.168.25.8
IP maska:	255.255.0.0	IP maska:	255.255.0.0

1.7 Protokol EPSNet UDP

Jak je na první pohled patrné, protokol EPSNet UDP vychází z charakteristik klasického UDP. Číslo portu je na straně PLC TECOMAT v režimu PC defaultně 61682. Režim PLC používá číslo portu 61681. Data v UDP paketu zasílaném do PLC začínají záhlavím délky 6 bytů a následuje vlastní zpráva, která má shodnou strukturu jako standardní protokol EPSNet. V jednom takovémto paketu lze najednou poslat až 5 EPSNet zpráv najednou a tím uspořít objem zasílaných dat. Každá takováto zpráva je číselně označena, aby nedošlo ke ztrátě. PLC tyto zprávy zpracovává v pořadí v jakém dorazily, protokol UDP ovšem není spojovaná služba a nikterak nekontroluje posloupnost paketů, může tedy nastat situace kdy PLC dostane novější zprávu rychleji než zprávu vyslanou dříve. Odpověď od PLC má stejný formát. Za záhlavím délky 6 bytů následuje příslušný počet odpovědí v protokolu EPSNet. Logická adresa stanice může být v rozsahu od 0 do 126.

1.7.1 EPSNet UDP datagram

EPSNet datagram do sebe navrství až 5 EPSNet zpráv s maximální velikostí 520 bytů. Hlavička datagramu je rozdělena do 4 částí a celkem zabírá 6 bytů. Tato hlavička je při přeposílání na sériový port v Translátoru odtržena. Význam jednotlivých částí podle [8] je následující:

byte 0 a 1 **MESI** – číslo relace

Slouží k identifikaci jednotlivých zpráv. PLC v odpovědi vrací totéž číslo relace, jaké nesl dotaz vyslaný z nadřazeného systému. Tak lze jednoznačně určit, ke kterému dotazu patří přijatá odpověď.

byte 2 **PN** – kód režimu komunikace

Určuje v jakém režimu se mají následující data zpracovat.

- 2 - režim **PC** (EPSNet slave)
- 3 - režim **PLC** (EPSNet multimaster)

byte 3 **R** – rezerva

byte 4 a 5 **DPLEN** – délka následujících dat

Určuje délku následujících dat (všechny zprávy EPSNet v tomto UDP paketu bez záhlaví). Byte 4 obsahuje vyšší byte délky, byte 5 obsahuje nižší byte délky

0	1	2	3	4	5	6...
MESI	PN	R	DPLEN	EPSNET		EPSNET
EPSNET				EPSNET		EPSNET

Obr. 1.7.1.1: Struktura EPSNet UDP

Celková délka dat v UDP paketu musí být povinně sudá. Pokud při komunikaci vychází lichá délka dat, pak je nutno na konec dat přidat jeden nulový byte a zvýšit celkovou délku dat v UDP paketu. Uvedená úprava délky je potřebná pouze pro vysílání UDP paketu, položka DPLEN v záhlaví se neupravuje (může být i lichá).

1.8 Komunikace PC a PLC v odlišných sítích

Pokud je počítač přiřazen do jiné sítě než PLC, musíme u obou nastavit IP adresu a výchozí bránu sítě. Každá síť má svoji specifickou adresu brány. Odlišnou síť určíme podle tříd IP adres. Třída IP adresy je určena 1. bytem IP adresy. Celkem známe 5 tříd, ale třídy s prvním bytem 224 a větším jsou vyhrazeny pro multicast a rezervu.

Tab. 1.8.1: Rozdělení adres sítě podle třídy IP

Třída adresy	1. byte IP adresy	2. byte IP adresy	3. byte IP adresy	4. byte IP adresy
A - 1 až 127	adresa sítě		adresa počítače	
B - 128 až 191	adresa sítě		adresa počítače	
C - 192 až 223	adresa sítě			adresa počítače

Tab. 1.8.2: Ukázka zapojení v odlišné síti

	PC	PLC
IP adresa:	192.168.12.3	192.168.5.9
IP maska:	255.255.255.0	255.255.255.0
Brána:	192.168.12.1	192.168.5.1

Podle Teco a.s. [8] jsou zprávy mezi sítěmi směrovány přes bránu, musí se v PLC nastavit její IP adresa. Nastavení adresy brány lze provést dvěma způsoby. První je přes vývojové prostředí Mosaic a druhé přímým zadáním na centrální jednotce. V prostředí Mosaic pomocí manažeru projektu v nastavení parametrů kanálů centrální jednotky do parametru „Výchozí brána“. Následně je třeba nastavené hodnoty zapsat do paměti v PLC tlačítkem „Uložit do PLC“. Pro nastavení přímo v centrální jednotce je třeba vstoupit do nastavovacího režimu a hodnoty výchozí brány zapsat do parametrů GW1 až GW4.

2. Protokol TCP/IP

Na Internetu se využívá mnoho typů protokolů ať už přímo TCP/IP nebo dřívějších protokolů ISO a ITU. Název TCP/IP vychází ze dvou jmen nejpoužívanějších protokolů v této rodině. Je zde rozebrán především síťový protokol IP a transportní vrstva UDP. Obě tyto vrstvy jsou v Translátoru využívány k přenosu EPSNet zpráv mezi cílovými počítači.

Architektura TCP/IP je členěna do čtyř vrstev. Funguje zde hierarchie „velení“. Vrstva je podřízena vrstvě o jednu vyšší a využívá vrstvu o jednu nižší k pokračování komunikace. Tyto vrstvy nevnímají komunikaci mezi dvěma počítači na nižších vrstvách a nejsou jí nijak ovlivněny.

1. Aplikační vrstva

Aplikační protokoly přenášejí data mezi dvěma aplikacemi a určují formát v jakém jsou data uspořádána. Narozdíl od ISO/OSI komunikuje přímo s transportní vrstvou. V tomto případě je aplikační protokol EPSNet UDP. Jedna strana funguje jako server a druhá jako klient. Je to takzvaná nejvyšší vrstva.

2. TCP/UDP vrstva

Tato vrstva zabezpečuje spojení mezi aplikacemi počítačů pomocí portů. Porty nabývají rozsahu od 0 do 65535, porty jsou navíc od sebe odděleny na UDP a TCP. Protokol TCP je spojovanou službou, tzn. kontroluje zda byla data doručena a v případě chyby je vyšle znova. Protokol UDP je nespojovanou službou, tzn. doručení dat nekontroluje a zcela přenechává tuto funkci aplikačnímu protokolu.

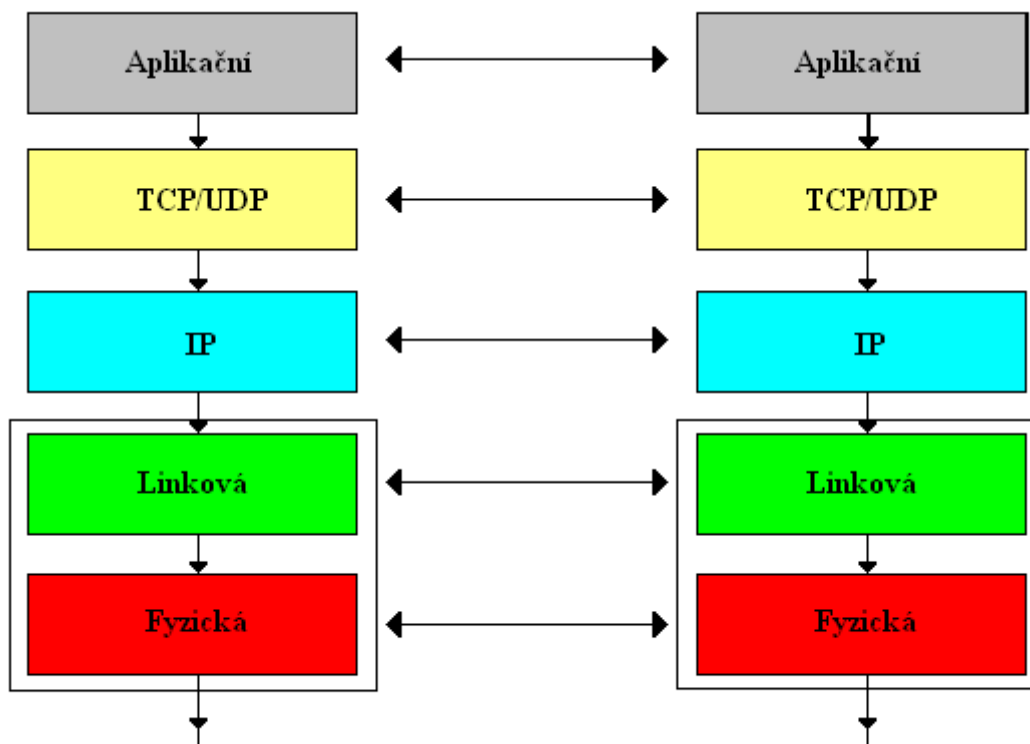
3. IP vrstva

IP vrstva zabezpečuje přenos dat mezi počítači umístěnými v internetu. Určující hodnotou je IP adresa počítače v síti Internet. Zajišťuje nespolehlivou datagramovou službu vzhledem k nespojovanému charakteru.

4. Linková a fyzická vrstva

Linkový protokol realizuje komunikaci mezi jednotlivými objekty uvnitř sítě například přes Ethernet a zároveň řeší převody na jiný druh linky třeba na Fast Ethernet.

Fyzická vrstva zapříčiňuje samotný fyzický přenos dat, jak přenést logickou 1 a 0, tvary konektorů, synchronizaci, přenosovou rychlost, typ přenosového média (optické vlákno, kroucená dvojlinka). TCP/IP tyto dva protokoly příliš neřeší a setkáváme se většinou s protokoly z modelu ISO OSI.



Obr. 2.1: Průchod dat přes jednotlivé vrstvy modelu TCP/IP

2.1 Protokol UDP – User Datagram Protocol

Protokol UDP je oproti protokolu IP protokolem vyšší vrstvy. Protokol IP přenáší data mezi počítači v internetu, zatímco protokol UDP určuje kterým aplikacím jsou data přiřazena pomocí portů. Protokoly TCP a UDP mají každý svoje vlastní porty i když číselné hodnoty jsou stejné, proto je možné vidět označení např. port 643/tcp a port 643/udp.

Spojovaná a nespojovaná služba:

Spojovaná

Pracuje na obdobném principu jako telefonní systém. Jeden účastník vytvoří spojení k danému cíli, ten může spojení buď přijmout nebo zamítnout. Při otevření tohoto informačního kanálu zůstává spojení aktivní po celou dobu až do zrušení jednou ze stran. Informaci odeslanou do tohoto tunelu si vybere účastník na druhé

straně a naopak. TCP ovšem kontroluje pořadí daných paketů a zda byly doručeny. V situaci kdy nedostane vysílací strana potvrzení o doručení, opět nepotvrzená data vyšle, zatímco zbytek dat se hromadí v paměti.

Nespojovaná

Tato služba nevytváří žádné spojení mezi subjekty. Vysílá data na určité místo, ale již nekontroluje jejich doručení. Pokud aplikace vyžaduje kontrolu dat a pořadí, musí se o to postarat vlastními prostředky. Výborným příkladem nespojované služby je protokol UDP. Využití nacházíme například v aplikacích pracujících se streamováním videa a audia, kde je nutné udržet tok dat a jakékoli jejich hromadění nebo zpoždění je nepřijatelné.

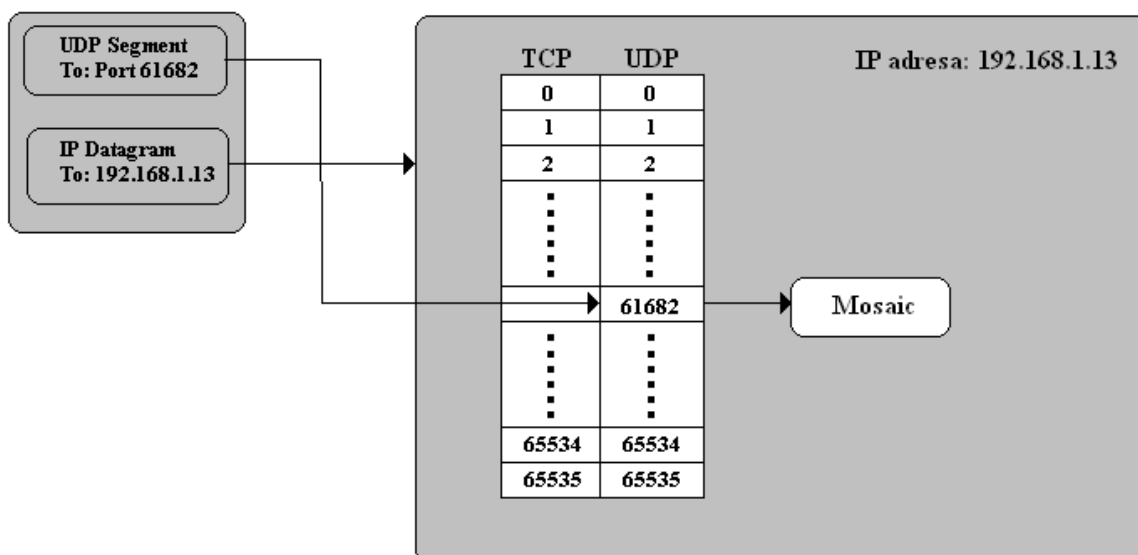
Porty

0 až 1023 – Serverové porty anebo označovány jako well known ports. Pro používání těchto portů je potřeba vyšších práv např. V systémech Linux práva uživatele „root“. Většinou jsou využívány systémem nebo důležitými aplikacemi.

1024 až 49151 – Registrované porty. Na těchto portech pracují různé nesystémové aplikace, šance na kolizi při zvolení jednoho z těchto portů je malá.

49152 až 65535 – Tzv. „klientské porty“. Jsou používány pro komunikaci klienta se serverem. Např. pro komunikaci s PLC se v protokolu EPSNet UDP v režimu PC používá defaultně port 61682.

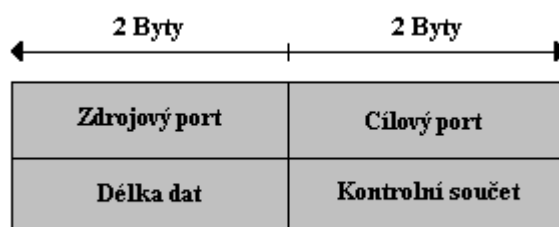
Rozdíl protokolu TCP a UDP je ten, že protokol UDP je nespojovanou službou. Odesílatel odešle UDP datagram a dále se nezajímá o jeho doručení. Tudíž se o doručení dat stará aplikační protokol, v tomto případě protokol EPSNet. Protokol UDP zároveň nekontroluje pořadí odesílaných datagramů, tudíž může nastat situace, kdy později vytvořená data jsou odeslána dříve a náprava závisí na aplikačním protokolu, v případě že pořadí je důležité.



Obr. 2.1.1: Příchozí data z Internetu na port UDP

2.1.1 Hlavička UDP

Hlavička UDP datagramu je rozdělena do 4 částí, každé o velikosti 2 bytů. Celková velikost hlavičky je tedy 8 bytů. Zdrojový port není nutné vyplňovat, toho se využívá v situacích kdy odesílatel neočekává další odpověď nebo končí komunikaci. Kontrolní součet lze také vynechat, ztrácí se tím ale možnost kontroly.



Obr. 2.1.1.1: Hlavička UDP datagramu

2.2 IP – InterNet Protocol

Je to protokol určený k přenosu dat mezi počítači připojených k Internetu. Každé síťové zařízení využívající protokol IP má svoje jednoznačně určující označení tzv. IP adresu, které mezi sebou nesmí kolidovat. IP adresa má velikost 4 bytů a je ve tvaru 192.168.1.1.

2.2.1 IP datagram

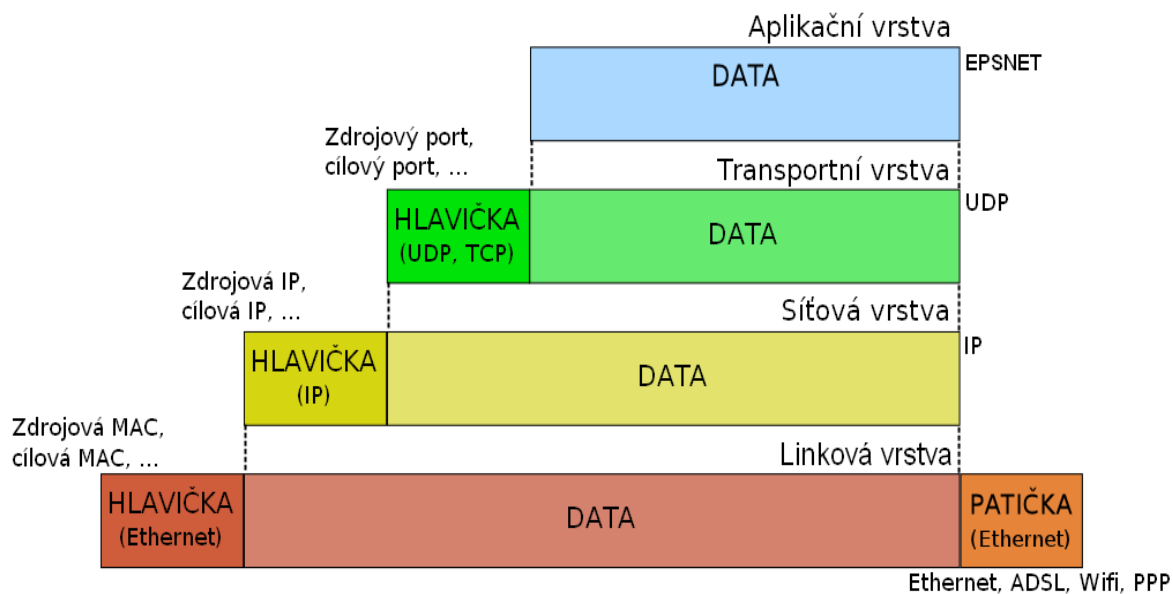
Nejdůležitější údaj je IP adresa odesilatele a příjemce. Samotné záhlaví datagramu má 20 bytů, může se ovšem zvětšit o velikost volitelných dat v hlavičce o dalších 40, celkově tedy až na 60 bytů.

0	8	16	31
Verze IP	Délka záhlaví	Typ služby	Délka IP datagramu
Identifikace IP datagramu		Příznaky	Posunutí fragmentu
Životnost	Protokol vyšší vrstvy	Kontrolní součet záhlaví	
IP adresa odesilatele			
IP adresa příjemce			
Volitelné položky záhlaví (0-320 bitů)			
Data (například UDP datagram)			

Obr. 2.2.1.1: Hlavička IP datagramu

2.3 Vrstvení EPSNet zprávy

Po vytvoření zprávy EPSNet dochází k několikanásobnému vrstvení hlaviček dat protokolů, potřebných k transportu zprávy z vývojového prostředí Mosaic po Internetu až do cílového PLC. Tato procedura však původní zprávu nijak neovlivňuje. Problémy se mohou vyskytnout při vzdáleném transportu přes internet, který ovlivňuje odezvu mezi PC a PLC, následovanou výpadky komunikace z nedostatečně rychlé odpovědi a ukončení spojení.



Obr. 2.3.1: Vrstvení hlaviček na zprávu EPSNet, podle vzoru [5]

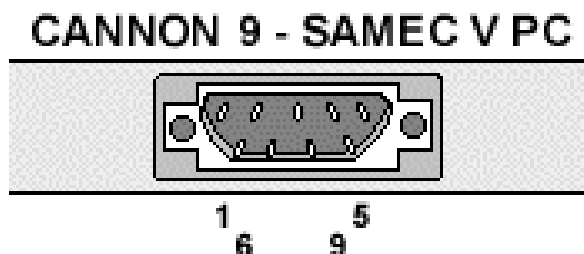
Na Obr. 2.3.1 lze pozorovat velikost původní EPSNet zprávy až do stavu při samotném fyzickém posílání a doprovázejícího obrovského nakupení dat.

3. Sériový port RS-232

Sériový port označovaný také názvem RS-232 se využívá jako komunikační rozhraní počítače a jiné elektroniky (v tomto případě PLC). Sériový port umožňuje sériovou komunikaci, to znamená že vysílá data po jednotlivých bitech v řadě za sebou na jednom vodiči. Jak lze vidět na obr. 3.1, číslování kolíků začíná v levém horním rohu jedničkou a končí v pravém dolním rohu devítkou. Označení a funkce jednotlivých kolíků jsou vysvětleny v tab. 3.1.

Tab. 3.1: Označení a funkce jednotlivých kolíků podle [9]

9-kolíkový konektor	25-kolíkový konektor	Označení	I/O
1	8	DCD - Detektor přijmaného signálu	Vstup
2	3	RxD - Přijmaná data	Vstup
3	2	TxD - Vysílaná data	Výstup
4	20	DTR - Pohotovost terminálu	Výstup
5	7	GND - Zem	---
6	6	DSR - Pohotovost vysílače	Vstup
7	4	RTS - Výzva k vysílání	Výstup
8	5	CTS - Pohotovost k vysílání	Vstup
9	22	RI - Indikátor volání	Vstup



Obr. 3.1: Číslování kolíků sériového portu podle [6]

Pro vytvoření funkčního spojení mezi oběma konci je zapotřebí zadat stejné hodnoty nastavení portu. V Translátoru jsou ponechány volby pro paritu, přenosovou rychlost a velikost stop bitu. Nastavení počtu data bitů je dáno pevně na 8, jelikož s tímto počtem bitů pracuje EPSNet protokol.

3.1 Parita

Parita je nejjednodušší způsob zabezpečení přenosu dat bez ohrožení výpočetní rychlosti. Ve vysílané zprávě se sečte počet jedničkových bitů a paritní bit doplní počet podle typu nastavené parity.

Žádná parita – Bez parity.

Sudá parita – Počet jedničkových bitů včetně paritního je sudé číslo.

Lichá parita – Počet jedničkových bitů včetně paritního je liché číslo.

Space parita – Tzv. nulová parita. Paritní bit je vždy log. 0, využíváný při komunikaci 7bitového zařízení s 8bitovým.

Mark parita – Paritní bit je vždy log. 1, využíváný při kompenzaci 7bitového provozu.

3.2 Baudrate

Baudrate je určení přenosové rychlosti na sériovém portu. Vyjadřuje počet změn přenosového signálu mezi log.0 a log.1. Do jedné změny signálu lze zakódovat i víc než jeden bit a proto nelze zaměnit označení bit za sekundu s jedním baudem.

Hodnoty baudrate se pohybují od 1200 do 115200. Zvolená rychlost je také závislá na délce propojovacího kabelu. Čím větší vzdálenost, tím nižší možný baudrate. V tabulce 3.2 se lze podívat na hodnoty baudrate vzhledem k délce kabelu při zachování standardu přenosu. Je nutné brát v potaz, že údaje byly naměřeny v laboratorním prostředí společnosti Texas Instruments a v reálném prostředí nelze těchto výsledků dosáhnout. Dále je z této rychlosti třeba odečíst změny využitě k odeslání start bitu a synchronizaci spojení pomocí stop bitu, pohybující se kolem 20%.

Tab. 3.2.1: Naměřené rychlosti sériového přenosu vzhledem k vzdálenosti

Baudrate	Vzdálenost v metrech
19200	15
9600	150
4800	300
2400	900

3.3 Stop bit

Stop bit je přidán na konec každého vyslaného rámce. Logická hodnota stop bitu je vždy log. 1. Nastavení stop bitu je délka jeho vysílání od velikosti 1 po 1,5 až 2 délky bitového intervalu. Během příjmu tohoto bitu zařízení zpracovává přijatý rámec. Pro většinu zařízení se ponechává délka stop bitu na velikosti 1.

4. Program Translátor

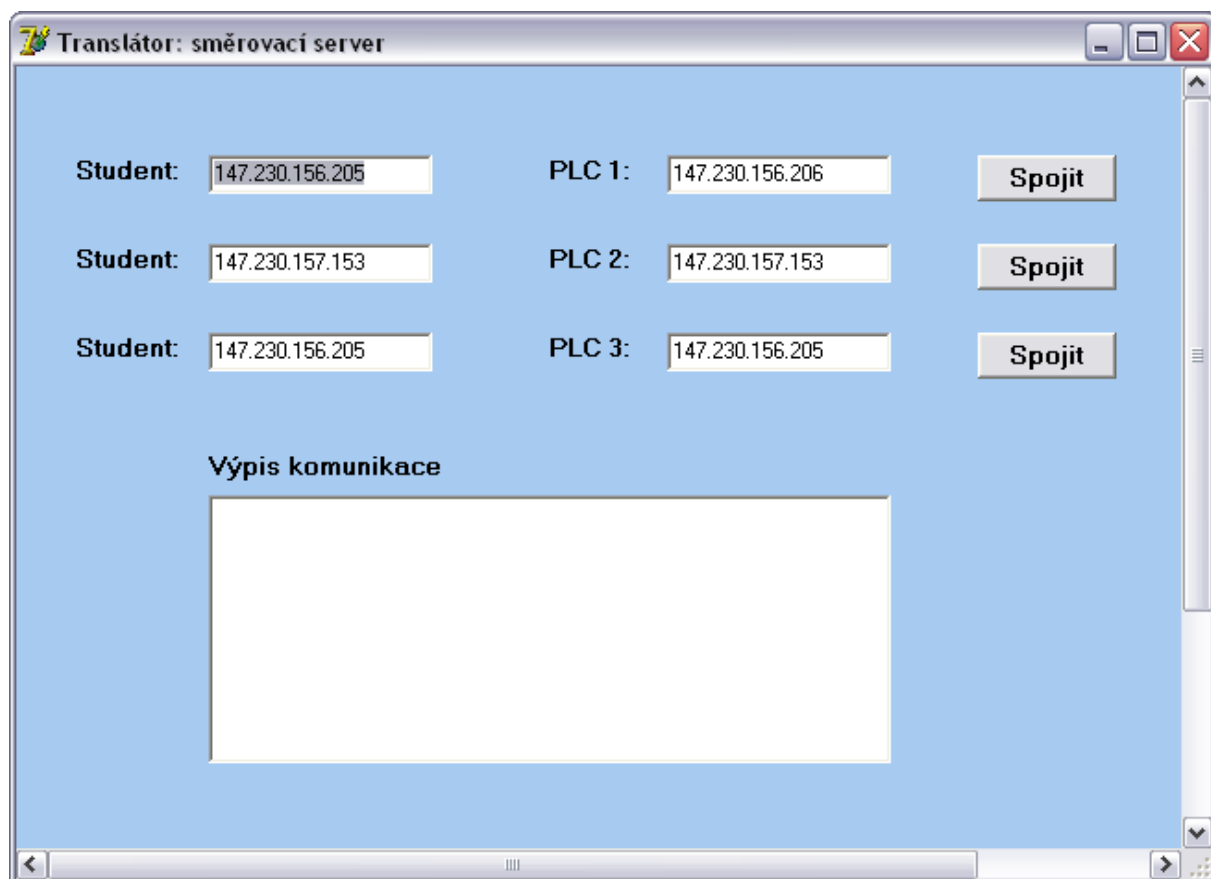
Program byl navrhnut pro praktické využití v laboratoři TK3. Zde byl vznesen požadavek na možnost připojení všech PC k daným PLC v učebně, které jsou starších modelů a nemusejí mít ethernetovou přípojku a jsou připojeny přes sériový port. Řídící a zároveň směrovací část je spuštěna u vyučujícího, sekundární část pak na počítači s automatem. Z toho vyplývá, že pro využívání jednotlivých PLC již student nemusí u tohoto počítače sedět, ale lze ho ovládat od jiného.

4.1 Princip fungování programu

Program je rozdělen do dvou aplikací. První z nich je umístěna na počítači, přes který se směřuje veškerý provoz od studentů k cílovým počítačům s automatem a zpět. IP adresy obou typů účastníku lze měnit podle aktuální potřeby, tudíž zde existuje variabilita a volnost v přesouvání jak automatů tak studentů. Druhá aplikace je spuštěna na počítači s automatem PLC zapojeným přes sériový port. Tato aplikace přijímá UDP data přes TCP/IP od směrovací stanice, přeposílá je na sériový port a zároveň ze sériového portu přes TCP/IP zpět. Obě aplikace jsou serverového typu, sekundární aplikaci lze ale označit za klientskou.

4.2 Translátor: směrovací aplikace

Tato aplikace má za úkol výměnu dat pomocí protokolu UDP mezi cílovými počítači. Obsahuje 6 editačních polí určujících IP adresy počítačů studentů a počítačů s připojeným fyzikálním modelem přes PLC. Hlavní síťovou součástí je UDP server INDY jako nativní balíček komponent pro vývojové prostředí Delphi. Způsob směrování je vyřešen odlišováním IP adres a portů, na které data přicházejí. Po spuštění komunikace se zablokují porty 61600, 61602 a 61604, každý pro jedno ze tří spojení, a program začne naslouchat pro příchozí data. Po příchozím spojení se podle čísla portu vybere program jako cíl adresu zadanou v opačném editboxu a následně ji odešle. Pro jednodušší manipulaci se transferovaná data přetypují na pole bytů. Pomocí informačního pole jsou průchozí data zobrazena v desítkové soustavě.



Obr. 4.2.1: Rozhraní serveru

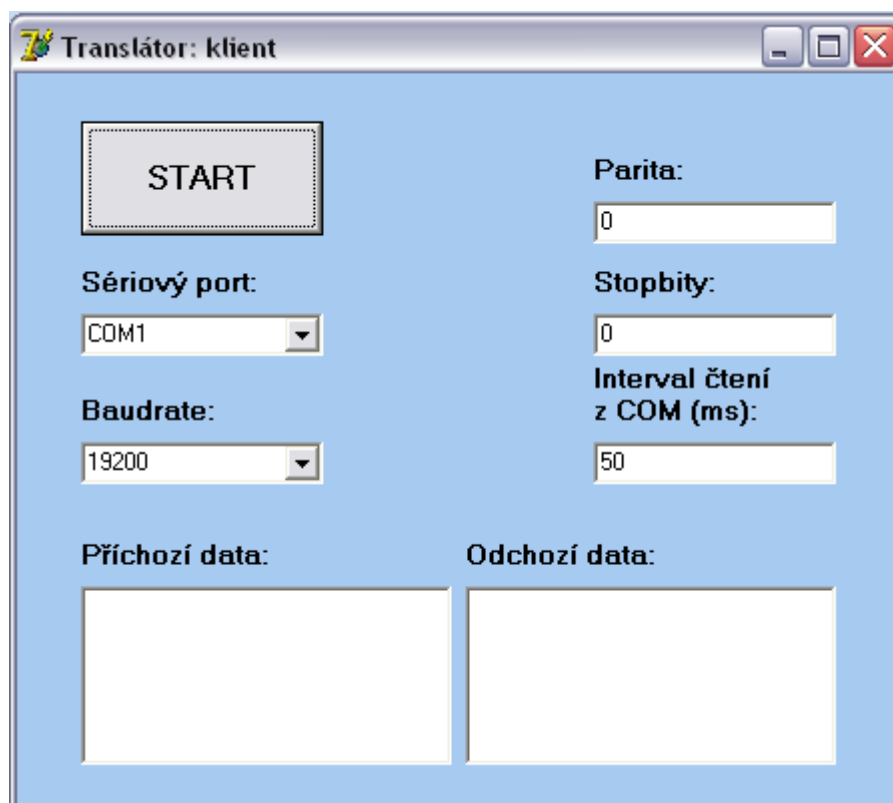
4.3 Translátor: klient

Tato aplikace je určena k zprostředkování PLC do internetu. Nastavení rychlosti přenosu, čísla portů, stop bitů a parity probíhá pomocí rozhraní programu. IP adresa serveru se nezadá, je automaticky určena adresou z příchozích dat. Přístup k sériovému portu je zpracován přímo přes rozhraní API. Pro lepší zpracování se přijatá data přes UDP ukládají opět do pole bytů. Následně se kvůli nemožnosti PLC zpracovávat EPSNet UDP protokol provede překlad na EPSNet zprávy. Od paketu se odstraní hlavička EPSNet UDP a do sériového portu se vyšle samostatná EPSNet zpráva. Po obdržení odpovědi od automatu se ke zprávě zpět přidá odstraněná hlavička ovšem s přepočítanou velikostí na aktuální hodnotu a pomocí UDP odešle na směrovací server.

Hodnoty parity a stop bitů v aplikaci:

Parita: 0=žádná, 1=lichá, 2=sudá, 3=mark, 4=space

Stop bity: 0=1, 1=1.5, 2=2



Obr. 4.3.1: Rozhraní klientské aplikace

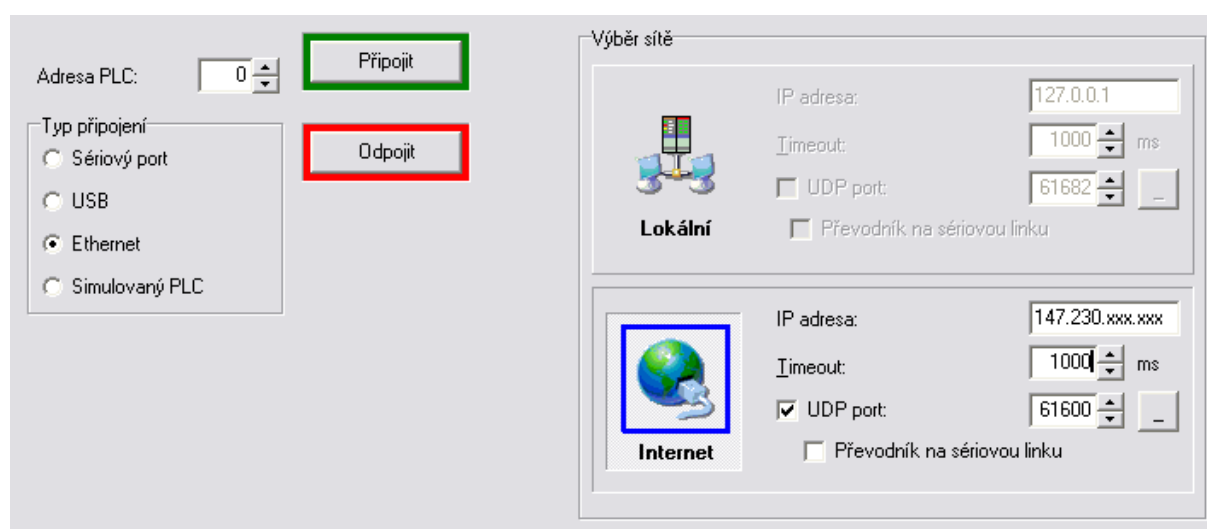
4.4 Spuštění Translátoru a vytvoření spojení

1. Vybereme počítač, na kterém bude spuštěný směrovací server Translátoru. V tomto počítači zajistíme průchodnost spojení vypnutím firewallu nebo jeho správným nastavením.
2. Spustíme směrovací server. V kolonkách studentů vyplníme IP adresy počítačů, z kterých se studenti připojují. Ke každému studentovi je na stejném řádku potřeba vyplnit IP adresu počítače s PLC. Po vyplnění dvojice klikneme na „Spojit“.
3. Na počítači s PLC spustíme klientskou část Translátoru. Zde vybereme sériový port a nastavíme jeho hodnoty shodně s nastavením portu u PLC. Klikneme na „Start“ aktivujeme UDP server a zároveň sériový port.
4. Zbývá už pouze připojení počítače s prostředím Mosaic. V manažeru projektu nastavíme typ připojení na Ethernet. Do cílové adresy vyplníme adresu

směrovacího serveru. UDP port vyplníme podle požadovaného PLC viz Tab 4.4.1. Zaškrtneme volbu odesílání pouze jedné EPSNet zprávy. Tlačítkem „Připojit“ vznikne funkční spojení se vzdáleným PLC.

Tab. 4.4.1: Dvojice portů s PLC

Port	Zařízení
61600	PLC 1
61602	PLC 2
61604	PLC 3



Obr. 4.4.1: Připojení v prostředí Mosaic

4.5 Průběh vývoje

Při prvním testování aplikace se objevily problémy v podobě zápisu dat na sériový port. Bylo využíváno komponenty Varian Async32, která má za úkol zprostředkovávat práci a komunikaci s portem bez větších znalostí vývojáře o API. Tato komponenta nesplnila očekávání a byla nahrazena ovládáním portu přímo přes API.

Další problém se vyskytnul při posílání dat z UDP serveru a portem. Zpráva EPSNet ze serveru je zabalena v EPSNet UDP datagramu, přičemž PLC zapojené přes sériový port tuto metodu nepodporuje. To bylo vyřešeno odtrhnutím hlavičky od zprávy, zpětně pak připsáním stejné hlavičky s přepočítanou velikostí dat v paketu.

4.6 Zásady používání

První podmínka co musí být splněna je nastavení Mosaicu na odesílání pouze jedné zprávy v EPSNet UDP paketu. Zde by nastaly potíže kvůli nemožnosti PLC ukládat zprávy do fronty a jejich dalším zpracováním. Pokud by tato situace byla ošetřena v programu, mohlo by dojít k zacyklení zpráv, tzn. i když by Mosaic vyslal nový dotaz, paketu by bylo na cestě zpět z PLC přiřazeno špatné číslo relace.

Druhá podmínka je správné zvolení intervalu čtení ze sériového portu. Moc nízký interval čtení způsobí, že program přečte zprávu ještě dříve než byla kompletně přenesena do počítače a odešle ji po více částech. Každá z nich má stejné číslo relace, takže Mosaic další zprávy se stejnou identifikací odhodí. Pokud se nastaví příliš vysoká hodnota, dochází při čtení k nakupení více zpráv do jedné. Ideální nastavení je takové, když je zde ponechána dostatečná rezerva k přečtení celé zprávy, ale zároveň je menší než vypršení času Mosaicu pro opakované vyslání dotazu. Ke snížení rizika kolapsu by bylo vhodné dané PLC nastavit na vyšší sériovou komunikační rychlost. Při testování se osvědčil interval 100 ms při baudrate 19200, finální odezva se pak pohybovala kolem 200 ms. Obrázek dosaženého spojení je v příloze A.

4.6.1 Diagram vytvořené sítě

Síť může být vytvořena několika způsoby propojení.

1. Počítač s prostředím Mosaic je na stejném počítači jako reálné PLC
2. Počítač s prostředím Mosaic je stejné místností jako reálné PLC, ale na jiném počítači.
3. Počítač s prostředím Mosaic je za bránou a v jiné síti než reálné PLC.

Na všech účastnících se počítačích musí být vypnut firewall nebo jiným způsobem ošetřena průchodnost spojení. Další podmínkou je vlastnictví veřejné IP adresy všemi účastníky. Obrázek možností připojení z různých lokalit je uveden v příloze B.

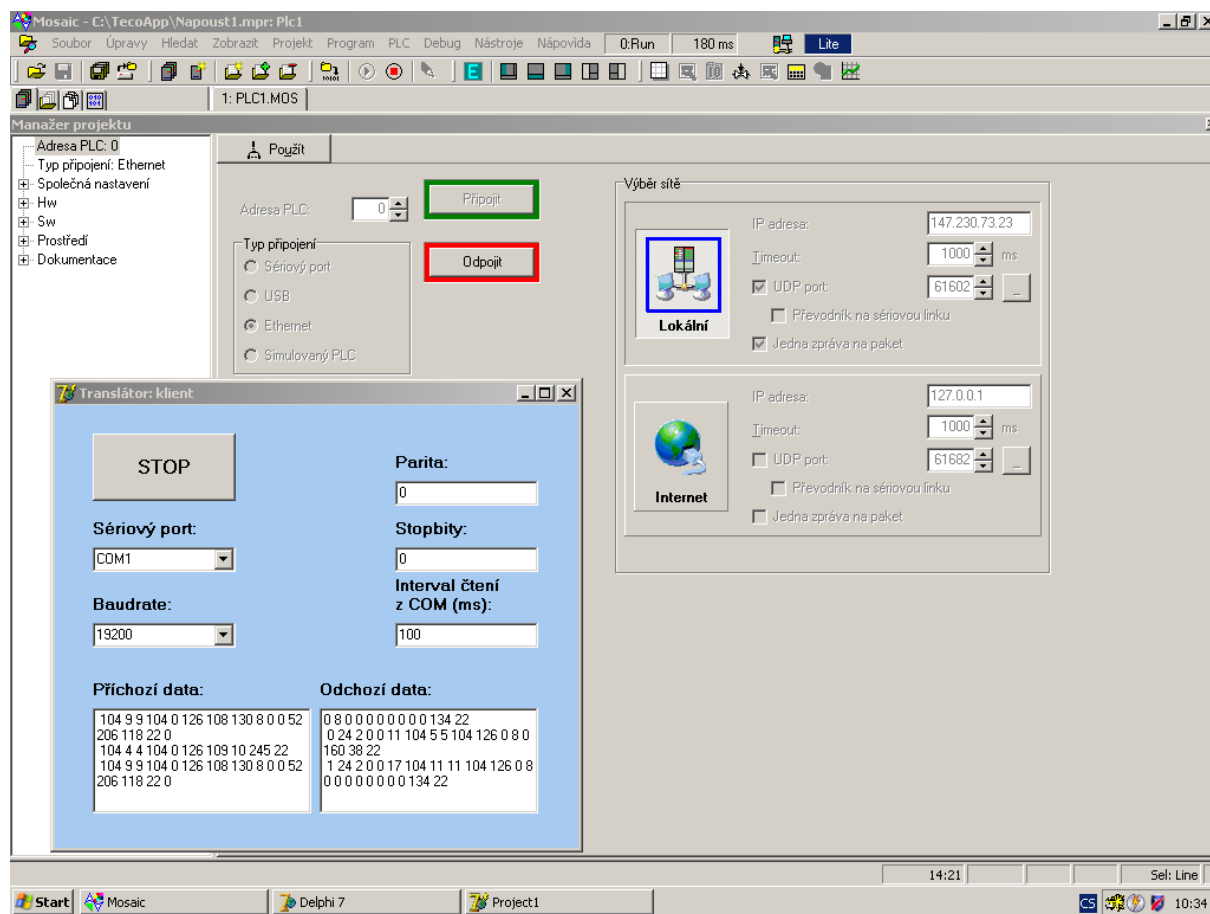
Závěr

Bakalářská práce je rozložena do čtyř logických celků. Nejprve je rozebrána problematika sítí a módů na nichž PLC pracují. Podrobně je zde popsán protokol EPSNet včetně principu jeho fungování, režimu komunikace a jiné. Některé zde popsané systémové příkazy byly během práce využívány pro přímou kontrolu navázání spojení. Samotná struktura paketu byla využita při čtení zprávy ze sériového portu. Znalost struktury EPSNet UDP byla zapotřebí při jejím odstraňování před vysláním do sériového portu. Druhá kapitola je věnována letmému pohledu na protokol TCP/IP se zaměřením na vrstvu UDP a IP, ve kterých se spojení odehrávalo. Kapitola třetí popisuje sériový port a nastavení jeho specifických vlastností, které jsou důležité pro vytvoření spojení. V poslední kapitole je popsáno samotné fungování programu, popis jeho dvou částí, návod pro obsluhu programu a zásady používání ke zdárnému propojení.

Výsledkem práce je podle mého názoru univerzální síťový nástroj pro propojení softwaru Mosaic se vzdáleným automatem připojeným přes sériový port. Tyto počítače mohou být i v odlišných sítích v rámci Internetu, musí ovšem splňovat požadavek na veřejnou IP adresu. Během vývoje prošel program několika stádií testování až do finální kompletně funkční podoby. Testování probíhalo v laboratoři TK3 na reálných automatech, kde bylo dosaženo pozitivních výsledků a úspěšného spojení. Obrázek s funkčním propojením je v příloze A.

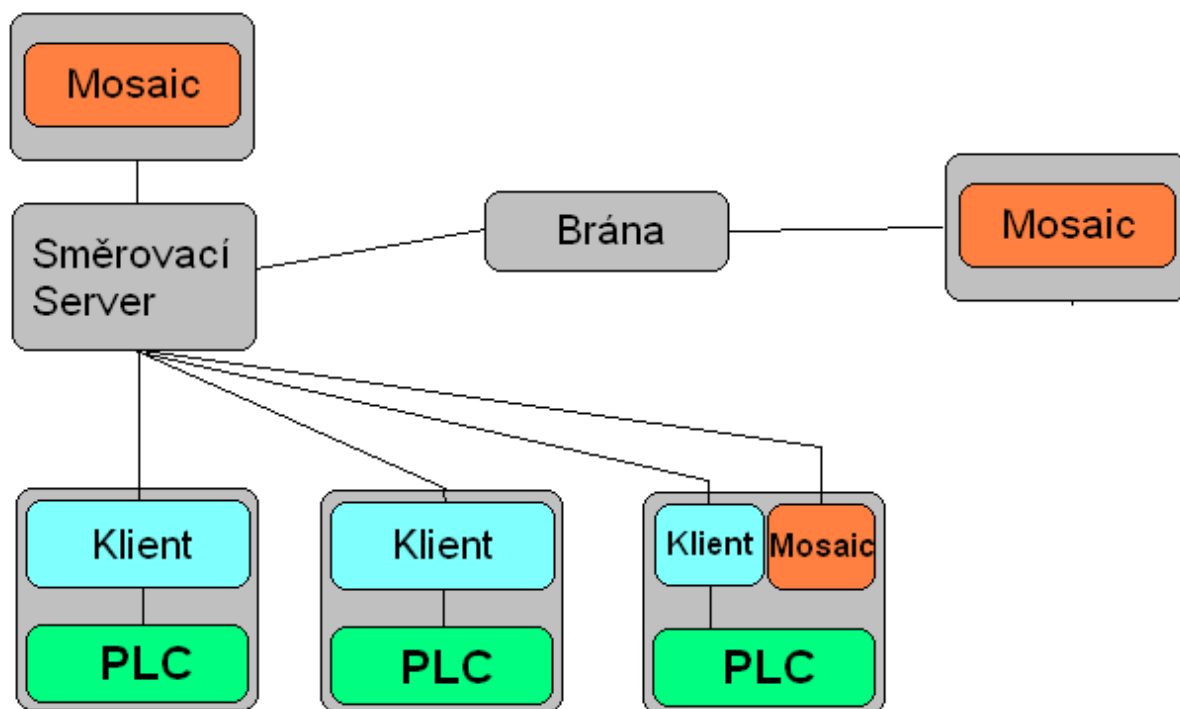
Zdrojový kód a spustitelný program jsou umístěny na přiloženém CD.

Příloha A



Obr. příloha A: Vytvořené spojení

Příloha B



Obr. příloha B: Diagram vytvořené sítě

Seznam použité literatury

- [1] Boyd, Tom: *Delphi: RS-232, serial comms, COM1, COM2* [online].
URL: <<http://sheepdogguides.com/dt5f.htm>>
- [2] Cantú, M: *Myslíme v jazyku Delphi 6*. Computer Press, a.s, Praha 2003
- [3] Clemenzi, Robert: *Serial I/O (RS232) Using Delphi (RS232)* [online] . Capital Personal Computer User Group, Inc
URL:<http://mc-computing.com/languages/Delphi/Delphi_SerialIO.htm#Delphi_References>
- [4] Dostálek, Libor-Kabelová, Alena: *Velký průvodce protokoly TCP/IP a systémem DNS*. Computer Press, a.s., Brno 2008. 488 stran
- [5] Mudrák, David: *Schéma zapouzdření aplikačních dat na jednotlivých vrstvách rodiny protokolů TCP/IP*[online].
URL: <http://cs.wikipedia.org/wiki/Soubor:Tcpip_zapouzdeni.svg>
- [6] Olmr, Vít: *HW server představuje - Sériová linka RS-232*[online]. HW server s.r.o., 12. Prosinec 2005.
URL: <<http://hw.cz/rs-232#konektory>>
- [7] Perry, Jason "Wedge" : *TSerialPort: Basic Serial Communications In Delphi* [online]. Yahoo! Inc. , 16 březen 1997.
URL: <http://www.geocities.com/macrotech_tr/hardware/tserialport.html>
- [8] Teco a.s.:*Sériová komunikace programovatelných automatů Tecomat - model 32 bitů* [online]. TECO a.s., 14. vydání, srpen 2008.
URL: <<http://tecomat.cz/docs/cze/general/txv00403.pdf>>.
- [9] Vykopal, Jan: *Sériové rozhraní v Delphi (1. díl)* [online]. Grafika Publishing, s.r.o., 20.11. 2001.
URL: <<http://www.builder.cz/art/delphi/delphiser.html>>